

Inteligencia artificial

Sistemas basados en el conocimiento

Gispert Sánchez, Francesc-Xavier

Lorente García, Ester

Martín Alaminos, David

31 de mayo de 2015

Índice

1. Introducción y descripción del problema	2
2. Modelización	4
2.1. Problema básico	4
2.2. Primera extensión	4
2.3. Segunda extensión	4
2.4. Tercera extensión	4
2.5. Cuarta extensión	4
3. Implementación	5
4. Resultados	6
4.1. Problema básico	6
4.2. Primera extensión	6
4.3. Segunda extensión	6
4.4. Tercera extensión	6
4.5. Cuarta extensión	6
4.6. Evolución respecto al tamaño de la entrada	6
5. Conclusión	7

1. Introducción y descripción del problema

Algunos problemas de optimización resultan computacionalmente intratables con los algoritmos exactos que se conocen a causa de su complejidad algorítmica, debida a la magnitud de la estructura combinatoria subyacente. Por ello, la inteligencia artificial utiliza técnicas que podan el espacio de búsqueda y lo inspeccionan guiadas por heurísticas para poder ofrecer soluciones aproximadas de forma sencilla.

Una familia de problemas cuya resolución se ha simplificado mucho con herramientas genéricas que automatizan este proceso de búsqueda es la de los problemas de planificación. El objetivo de estos problemas es construir un plan o secuencia de acciones que permitan llegar a un cierto objetivo partiendo de un estado inicial. Se trata, pues, de problemas de síntesis, en los que hay que construir una solución basándose en el conocimiento del dominio (en este caso, en el estado del entorno, las acciones disponibles en cada momento y el objetivo al que se desea llegar).

La comunidad de inteligencia artificial ha desarrollado, a lo largo de los años, herramientas capaces de resolver problemas de planificación independientemente del dominio. Así, el usuario de estas herramientas solo tiene que modelizar su dominio y su problema mediante un lenguaje formal de representación y el sistema se encarga de la resolución automática del problema.

En este trabajo, lo ejemplificamos mediante la resolución de un problema simple de asignación que se puede modelizar como un problema de planificación para guiar adecuadamente la construcción de la asignación. El problema consiste en la distribución de T tareas de programación entre P programadores. Cada tarea tiene asociada un nivel de dificultad (1, 2 o 3, por orden creciente) y un tiempo estimado de desarrollo (un número de horas); por su lado, cada programador tiene asociada un nivel de habilidad (1, 2 o 3, por orden creciente) y una categoría según su calidad (1 o 2, por orden decreciente). Un programador puede resolver una tarea del mismo nivel de dificultad que su nivel de habilidad en el tiempo establecido o bien una tarea con un nivel de dificultad inmediatamente superior a su habilidad añadiendo dos horas adicionales. Tras la resolución de una tarea, se crea una tarea de revisión de una hora si el programador era de primera calidad o de dos horas en caso contrario, independientemente de si el programador que la lleva a cabo tiene una habilidad inferior; esta tarea de revisión tiene la misma dificultad que la inicial y tiene que ser llevada a cabo por otro programador distinto.

En esta práctica hemos considerado la versión básica del problema y las cua-

tro extensiones propuestas, de forma incremental. El problema básico consiste en la resolución directa del problema de asignación sin tener en cuenta las tareas de revisión. En la primera extensión, se añaden estas tareas de revisión al problema. La segunda extensión pretende, además, minimizar el tiempo total de resolución de las tareas. En la tercera extensión se limita el número de tareas por programador a 2 para facilitar el trabajo en paralelo. Finalmente, en la cuarta extensión, se pretende minimizar el producto entre el número de personas con tareas asignadas y el tiempo total de resolución.

Se desarrolla, pues, un modelo expresado en PDDL para cada uno de los problemas considerados, analizando los elementos necesarios y las diferencias entre ellos, y se resuelven utilizando la herramienta de planificación Fast Forward. El desarrollo de la práctica, así como la estructura de este informe, está influenciado por la metodología de la ingeniería del conocimiento vista en la práctica anterior, con un desarrollo incremental y basada en distintos prototipos (aunque en este caso, debido a la simplicidad del problema, las fases de identificación del problema, conceptualización y formalización prácticamente se han unido en una sola).

En el resto de este documento, explicamos los pasos del desarrollo de nuestra solución y mostramos y analizamos los resultados obtenidos.

2. Modelización

En esta sección se describen detalladamente los elementos que aparecen en los modelos desarrollados para cada una de las versiones del problema consideradas. Como se explica en la Sección 3, el modelo para cada extensión se ha construido añadiendo funcionalidades al modelo para la extensión previa. Por lo tanto, y para evitar repeticiones innecesarias, para cada extensión solo se detallan los cambios introducidos.

2.1. Problema básico

2.2. Primera extensión

2.3. Segunda extensión

2.4. Tercera extensión

2.5. Cuarta extensión

3. Implementación

En la Sección 2, se ha descrito de forma detallada el modelo desarrollado para cada una de las variaciones del problema tratadas; estos modelos se encuentran implementados (expresados en el lenguaje PDDL) en el código adjunto a este documento. En esta sección, pues, se comenta el proceso seguido para su implementación.

Para implementar el modelo final, se ha optado por un esquema de diseño incremental basado en prototipos, siguiendo el guión propuesto en el enunciado de la práctica. Es decir, se empezó con una solución para el problema básico propuesto, que sirvió esencialmente como toma de contacto con el lenguaje PDDL y la herramienta Fast Forward, y esta se fue ampliando poco a poco y de forma ordenada para tener en cuenta cada una de las extensiones trabajadas. Por lo tanto, a lo largo del desarrollo de la práctica se han ido construyendo prototipos, todos ellos perfectamente funcionales, cada vez teniendo en cuenta más elementos del problema hasta llegar a la última extensión.

Sin embargo, la implementación del prototipo correspondiente a cada extensión partiendo de la anterior se ha llevado a cabo prácticamente en un solo paso (o sea, sin desarrollar diversos prototipos intermedios entre una extensión y la siguiente), puesto que la estructura en extensiones propuesta en el enunciado era suficientemente sencilla (es decir, los añadidos de una extensión respecto a la previa eran tan pocos que no justificaba una metodología más compleja).

Aparte de los modelos en PDDL, también se ha desarrollado un pequeño *script* en Python (se encuentra también en el código adjunto a este documento) para la generación aleatoria de instancias de entrada de nuestro problema y se ha utilizado para los experimentos descritos en la Subsección 4.6.

4. Resultados

4.1. Problema básico

4.2. Primera extensión

4.3. Segunda extensión

4.4. Tercera extensión

4.5. Cuarta extensión

4.6. Evolución respecto al tamaño de la entrada

5. Conclusión

En este trabajo hemos resuelto un problema práctico modelizándolo como problema de planificación y utilizando un sistema de planificación automática para su resolución. A pesar de tratarse de un problema relativamente simple, este problema nos ha permitido hacernos una idea del tipo de problemas a los que se pueden aplicar las técnicas de inteligencia artificial aprendidas y, en particular, de la capacidad expresiva de lenguajes formales como PDDL y la potencia de herramientas como Fast Forward.

En esta práctica, hemos analizado el problema de asignación de tareas de un proyecto a los programadores del equipo de desarrollo y hemos modelizado los elementos que intervienen en este problema mediante objetos, predicados y acciones de PDDL. Para ello, se ha seguido un desarrollo incremental guiado por las extensiones propuestas en el enunciado de la práctica.

De este modo, hemos podido utilizar el planificador Fast Forward para solucionar instancias del problema debidamente seleccionadas y estudiar los resultados obtenidos. Así, pues, hemos podido comprobar como los sistemas de planificación automáticos diseñados de forma genérica y sin conocimiento del dominio son capaces de ofrecer buenas aproximaciones en tiempos de ejecución muy razonables, a pesar de su facilidad de uso y su potencia expresiva.